Final Report on ONR Contract N00014-94-C-0007

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

Nimrod Megiddo

May 1997

The subject of research under this contract was topics in linear programming and related problems. The problems we have investigated are quite diverse.

1.

We investigated in [1] a theoretical framework for incorporating horizontal and vertical decomposition techniques into interior-point methods for linear programs. Corresponding to the linear program: Maximize $\boldsymbol{c}^T\boldsymbol{x}$ subject to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$, $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{b}$, ands $\boldsymbol{x} \geq \boldsymbol{0}$, we introduced two functions in the penalty parameter $t > 0$ and the Lagrange relaxation parameter vector $\boldsymbol{w}$, $\tilde{f}^p(t, \boldsymbol{w})$, the maximum of $\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{w}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{a}) + t\sum_{j=1}^{n} \ln x_j$ subject to $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{x} > \boldsymbol{0}$ (for horizontal decomposition), and $\tilde{f}^d(t, \boldsymbol{w})$, the minimum of $\boldsymbol{a}^T\boldsymbol{w} + \boldsymbol{b}^T\boldsymbol{y} - t\sum_{j=1}^{n} \ln z_j$ subject to $\boldsymbol{B}^T\boldsymbol{y} - \boldsymbol{z} = \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{w}$ and $\boldsymbol{z} > \boldsymbol{0}$ (for vertical decomposition). For each $t > 0$, $\tilde{f}^p(t, \cdot)$ and $\tilde{f}^d(t, \cdot)$ are strictly convex C^∞ functions with a common minimizer $\hat{\boldsymbol{w}}(t)$, which converges to an optimal Lagrange multiplier vector $\boldsymbol{w}^*$ associated with the constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$ as $t \to 0$, and enjoy the strong self-concordance property given by Nesterov and Nemirovsky. Based on these facts, we developed conceptual algorithms with the use of Newton's method for tracing the trajectory $\{\hat{\boldsymbol{w}}(t) : t > 0\}$, and analyzed their computational complexity.

2.

Vavasis and Ye proposed a layered-step primal-dual interior-point algorithm, whose number of operations has an upper bound that depends only on the coefficient matrix A and not on b and c. They defined the concept of crossover events for a linear programming problem that provides important insight into the behavior of the path of centers. The number of operations of the layered-step interior-point algorithm depends on the number of crossover events. Although the number depends on b and c, they prove that it is bounded by $(1/2)n(n-1)$. The question of whether there could be more than n crossover

19970523 151

DTIC QUALITY INSPECTED 1

events was left open in their paper. If one could prove that the number is bounded by $O(n)$, the complexity of the layered-step interior-point algorithm could be reduced by a factor of n. In [4] we presented a linear programming instance with more than $(1/8)n^2$ crossover events. We believe that the instance helps much for understanding the behavior of the path of centers.

The main drawback of the Vavasis-Ye algorithm is the use of an unknown big constant g. In [6] we proposed a simplified layered-step interior-point algorithm which did not use any unknown big constant. The complexity of the simplified algorithm is the same as that of Vavasis and Ye.

Like the algorithm of Tardos, the Vavasis-Ye algorithm solves flow problems in strongly polynomial time. It is called a "layered-step" interior-point algorithm, since it occasionally uses a layered least squares direction to compute a new iterate. It is at least as fast as the $O(\sqrt{n}L)$-iteration primal-dual path-following algorithms. Furthermore, if the path of centers has an almost straight part, the layered step may accelerate the algorithm. In particular, it attains an exact optimal solution when the iterate is close enough to a solution. So a layered-step interior-point algorithm is not only efficient in theory, but may also become a very good algorithm in practice.

The number of arithmetic operations performed by the algorithm is bounded in terms of a big constant $\bar{\chi}_A$ which is defined as the maximum of $\|A^T(ADA^T)^{-1}AD\|$, where D is a diagonal matrix whose diagonal entries are positive. This number is used for (i) computing the search direction, and (ii) constructing a problem which is equivalent to the original problem with a trivial initial primal-dual interior feasible solution when no feasible initial point is available. A drawback of the VY algorithm is that a good estimate of $\bar{\chi}_A$ should be known in advance, which may be difficult to compute. It may, however, be estimated by 2^L if A is an integral matrix of input size L.

We proposed a modification of the layered-step interior-point algorithm. Our algorithm does not use any unknown number for computing the search direction. Instead, we need an estimate of the norm of an optimal solution, but it is only necessary for constructing an equivalent problem to initiate the algorithm. If we know $\bar{\chi}_A$, we obtain a bound on the norm of an optimal solution as shown in by Vavasis and Ye and hence our algorithm is implementable.

3.

We made progress on the problem of solving two-person games represented as game trees. In [2] we presented new algorithms both for zero-sum games and general games. Our algorithms are, in general, exponentially better than the standard approach to the problem. They thus provide the first practical method for solving games that are not toy problems nor have a special structure.

We found a method that avoids the exponential blowup of the normal-form transformation. The basic idea is that the outcome of the game depends only on the distribution of probability weights that a randomized strategy induces on the leaves of the tree. We represent a strategy compactly in terms of these realization weights. These are defined directly in terms of the game tree, so their total size is linear rather than exponential in its size. This compact representation has a number of advantages. Using realization weights and LP duality, equilibrium strategies can then be found by solving a corresponding LP or LCP. We obtain the following two major results: 1. The optimal strategies of a two-player zero-sum perfect-recall game in extensive form are the solutions of a linear program whose size, in sparse representation, is linear in the size of the game tree. 2. The Nash equilibria of a general two-player perfect-recall game in extensive form are the solutions of a linear complementarity problem whose size, in sparse representation, is linear in the size of the game tree.

4.

Stochastic programming is a relatively young area of mathematical programming. Practitioners are now using stochastic models much more often than before because of the rapidly growing computational capabilities (including powerful hardware, modern software for linear programming, and high level modeling tools). The theoretical foundation, however, has not been established in a satisfactory way. Furthermore, the potential of using interior point algorithms has not been fully pursued. Our studies of decomposition for interior point algorithms are aimed at this.

Problems with stochastic parameters can be studied with various approaches and it is not yet clear what is the right one. There has been a growing interest among computer scientists in so-called online algorithms. These algorithms deal with multistage decision problems where data is revealed incrementally, while decisions have to be made. In stochastic programming one assumes some probability distribution over future parameters and has to make decisions in the present. Problems of this kind can often be posed as Markov decision processes or infinite horizon dynamic programming, and can sometimes be solved by linear programming (yet in a way different from what stochastic programming does). Recent work in stochastic programming suggests solving "simplified" sampled problems based on a set of possible scenarios. Another field, that is now called stochastic optimization, studies the question of searching for the optimum of a system whose performance is estimated by a stochastic simulation and the parameters can be chosen by a decision maker. In addition, there is the field of stochastic control that is supposed to deal with very similar questions. Moreover, people in AI now develop more methods for such problems using machine learning. We believe all of this raises very interesting foundational questions about the relationships among the various fields, and we need to test them on various basic problems. We began a study of the multi-period warehouse problem to see how sampled scenarios can help finding good inventory policies. This study is both theoretical and experimental, using a simulation model.

Infanger and Megiddo have laid the foundation for the dual value iteration method, a practical approach to optimizing linear systems. Our research in this area was motivated by the practical problem of multi-product production scheduling. Despite the vast literature on this problem, it seems that currently there is no good method for solving it in practice. Deterministic models (mainly multi-period linear programming) do not address the stochastic nature of the problem. Stochastic programming turns out to be too complicated. Dynamic programming and Markov decision processes have limited practical value due to the so-called curse of dimensionality. This is due to the need to discretize the problem and the number of discrete states of a system is typically quite large.

We consider a system whose *state* may change at discrete times $1, 2, \ldots$ The state is described by a *state vector* $\boldsymbol{s} = (s_1, \ldots, s_m)^T \in \Re^m$. Here we focus on a system whose state space is a bounded convex polyhedron given by a set of linear inequalities $\boldsymbol{M}\boldsymbol{s} \geq \boldsymbol{a}$ for some $\boldsymbol{M} \in \Re^{r\times m}$ and $\boldsymbol{a} \in \Re^r$. Depending on the *action* we take when the system is at state $\boldsymbol{s}$, the system undergoes a transition into a new state $\boldsymbol{s}'$. Below we consider stochastic systems, but let us, for simplicity, begin with deterministic ones. We first consider state transitions defined as follows. The actions are described by vectors $\boldsymbol{x} \in \Re^n$. They are subject to constraints dictated by a mechanism specified by matrices $\boldsymbol{C} \in \Re^{m\times n}$ and $\boldsymbol{B}, \boldsymbol{B}' \in \Re^{m\times m}$ and a vector $\boldsymbol{d} \in \Re^m$:

$$\boldsymbol{C}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{s} = \boldsymbol{B}'\boldsymbol{s}' + \boldsymbol{d} \ ,$$

where $\boldsymbol{B}'$ is nonsingular (so $\boldsymbol{s}'$ is determined by $\boldsymbol{s}$ and $\boldsymbol{x}$). Also, there are additional linear inequalities $\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}$ ($\boldsymbol{A} \in \Re^{p\times n}$, $\boldsymbol{b} \in \Re^p$) that $\boldsymbol{x}$ must satisfy. When we take action $\boldsymbol{x}$ we incur a *cost* of $\boldsymbol{c}^T\boldsymbol{x}$ ($\boldsymbol{c} \in \Re^n$). The actions are taken at times $1, 2, \ldots$, which we call *stages*, and the cost is subject to a *discount factor* of $\lambda < 1$ per stage. Suppose the initial state is $\boldsymbol{s}^0$, so $\boldsymbol{M}\boldsymbol{s}^0 \geq \boldsymbol{a}$. Suppose, by induction, we have chosen actions $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{k-1}$ and states $\boldsymbol{s}^1, \ldots, \boldsymbol{s}^{k-1}$, respectively, so that

$$\begin{aligned} \boldsymbol{C}\boldsymbol{x}^j + \boldsymbol{B}\boldsymbol{s}^{j-1} &= \boldsymbol{B}'\boldsymbol{s}^j + \boldsymbol{d} \\ \boldsymbol{M}\boldsymbol{s}^j &\geq \boldsymbol{a} \\ \boldsymbol{A}\boldsymbol{x}^j &\geq \boldsymbol{b} \end{aligned}$$

for $j = 1, \ldots, k-1$, and now choose an action $\boldsymbol{x}^k$ and a state $\boldsymbol{s}^k$ so that

$$\begin{aligned} \boldsymbol{C}\boldsymbol{x}^k + \boldsymbol{B}\boldsymbol{s}^{k-1} &= \boldsymbol{B}'\boldsymbol{s}^k + \boldsymbol{d} \\ \boldsymbol{M}\boldsymbol{s}^k &\geq \boldsymbol{a} \\ \boldsymbol{A}\boldsymbol{x}^k &\geq \boldsymbol{b} \ . \end{aligned}$$

Thus, we have defined by induction a *policy* (a feasible solution) for the "infinite-horizon" problem whose total discounted cost is $\sum_{k=0}^{\infty} \lambda^k x^{k+1}$. We wish to minimize this quantity. Note that the problem can be presented as an infinite linear programming problem.

Unlike the classical value iteration method that computes successive approximations for $L(s)$ from above, our method proceeds by computing successive approximations from *below*. An iteration of the algorithm begins with a piecewise linear function $L_k(\boldsymbol{s})$, given as the maximum of a finite number of linear functions:

$$\ell_i(\boldsymbol{s}) = (\boldsymbol{a}^i)^T \boldsymbol{s} + \delta_i \quad (i = 1, \ldots, k)$$

so

$$L_k(\boldsymbol{s}) \equiv \max_{1 \le i \le k} \{\ell_i(\boldsymbol{s})\} \le L(\boldsymbol{s}) \quad \text{for all } \boldsymbol{s} \ .$$

During the iteration, the algorithm picks some state $\boldsymbol{s}$ and computes actions at $\boldsymbol{s}$ that are optimal relative to the current approximation. This is accomplished by solving the following optimization problem:

$$\begin{aligned} \underset{(\boldsymbol{x},\boldsymbol{s}')}{\text{Minimize}} \quad & \boldsymbol{c}^T \boldsymbol{x} + \lambda L_k(\boldsymbol{s}') \\ \text{subject to} \quad & (\boldsymbol{x}, \boldsymbol{s}') \in \mathcal{F}(\boldsymbol{s}) \ . \end{aligned}$$

The latter can be solved as a linear programming problem by introducing an auxiliary variable t:

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{s}',t}{\text{Minimize}} \quad & \boldsymbol{c}^T \boldsymbol{x} + \lambda t \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}'\boldsymbol{s}' = -\boldsymbol{B}\boldsymbol{s} + \boldsymbol{b} \\ & \boldsymbol{M}\boldsymbol{s}' \ge \boldsymbol{a} \\ & \boldsymbol{D}\boldsymbol{x} \ge \boldsymbol{d} \\ & t - (\boldsymbol{a}^i)^T \boldsymbol{s}' \ge \delta_i \quad (i = 1, \ldots, k). \end{aligned}$$

Denote the optimal value of this problem by $LP_k(\boldsymbol{s})$. The dual problem is

$$\begin{aligned} \underset{\boldsymbol{y},\boldsymbol{z},\boldsymbol{w},u_1,\ldots,u_k}{\text{Maximize}} \quad & -(\boldsymbol{B}\boldsymbol{s})^T \boldsymbol{y} + \boldsymbol{b}^T \boldsymbol{y} + \boldsymbol{a}^T \boldsymbol{z} + \boldsymbol{d}^T \boldsymbol{w} + \sum_{i=1}^{k} \delta_i u_i \\ \text{subject to} \quad & \boldsymbol{A}^T \boldsymbol{y} + \boldsymbol{D}^T \boldsymbol{w} = \boldsymbol{c} \\ & -(\boldsymbol{B}')^T \boldsymbol{y} + \boldsymbol{M}^T \boldsymbol{z} - \sum_{i=1}^{k} u_i \boldsymbol{a}^i = \boldsymbol{0} \\ & \sum_{i=1}^{k} u_i = \lambda \\ & z, w, u_i \ge 0 \quad (i = 1, \ldots, k) \end{aligned}$$

Obviously, $LP_k(\boldsymbol{s})$ is a convex piecewise linear function of s, and $LP_k(\boldsymbol{s}) \le L(\boldsymbol{s})$ for all $\boldsymbol{s}$. In practice we cannot compute $LP_k(\boldsymbol{s})$ for all $\boldsymbol{s}$. Suppose we compute $LP_k(\hat{\boldsymbol{s}})$

for a certain state $\hat{\boldsymbol{s}}$ by solving the dual problem. Thus, we obtain optimal dual values $\boldsymbol{y} = \boldsymbol{y}(\hat{\boldsymbol{s}})$, $\boldsymbol{z} = \boldsymbol{z}(\hat{\boldsymbol{s}})$, $\boldsymbol{w} = \boldsymbol{w}(\hat{\boldsymbol{s}})$, and $u_i = u_i(\hat{\boldsymbol{s}})$ $(i = 1, \ldots, k)$. Let us denote

$$\boldsymbol{a}^{k+1} = \boldsymbol{a}^{k+1}(\hat{\boldsymbol{s}}) = -\boldsymbol{B}^T\boldsymbol{y}$$

and

$$\delta_{k+1} = \delta_{k+1}(\hat{\boldsymbol{s}}) = \boldsymbol{b}^T\boldsymbol{y} + \boldsymbol{a}^T\boldsymbol{z} + \boldsymbol{d}^T\boldsymbol{w} + \sum_{i=1}^{k} \delta_i \boldsymbol{u}_i \ .$$

Since the feasible domain of the dual problem does not depend on $\boldsymbol{s}$, it follows that the optimal dual solution computed for $\hat{\boldsymbol{s}}$ is feasible for *any* $\boldsymbol{s}$. It follows that for all $\boldsymbol{s}$,

$$\ell_{k+1}(\boldsymbol{s}) \equiv \ell_{k+1}(\boldsymbol{s}; \hat{\boldsymbol{s}}) \equiv (\boldsymbol{a}^{k+1}(\hat{\boldsymbol{s}}))^T\boldsymbol{s} + \delta_k(\hat{\boldsymbol{s}}) \leq LP_k(\boldsymbol{s}) \leq L(\boldsymbol{s}) \ .$$

Thus, the linear functional $\ell_{k+1}(\boldsymbol{s})$ can be used to update the piecewise linear approximation of $L(\boldsymbol{s})$:

$$L_{k+1}(\boldsymbol{s}) \equiv \max\{L_k(\boldsymbol{s}), \ell_{k+1}(\boldsymbol{s})\} \ .$$

Note that $\ell_{k+1}(\hat{\boldsymbol{s}}; \hat{\boldsymbol{s}}) = LP_k(\hat{\boldsymbol{s}})$.

As long as there exist states $\hat{\boldsymbol{s}}$ and $\boldsymbol{s}$ such that

$$\ell_{k+1}(\boldsymbol{s}; \hat{\boldsymbol{s}}) > L_k(\boldsymbol{s})$$

a better approximation $L_{k+1}(\boldsymbol{s})$ of $L(\boldsymbol{s})$ can be obtained. Otherwise, further iterations will produce the same function $L_k(\boldsymbol{s})$. The following theorem establishes that if this happens then an optimal solution has been found:

$$\text{If} \quad LP_k(\boldsymbol{s}) \leq L_k(\boldsymbol{s}) \quad \text{for all } \boldsymbol{s}$$

$$\text{then} \quad L_k(\boldsymbol{s}) = L(\boldsymbol{s}) \quad \text{for all } \boldsymbol{s} \ .$$

It is interesting to note that in order to establish the optimality of an action at $\boldsymbol{s}^0$ one does not need to consider all possible states.

5.

While studying the open area of online algorithms for linear programming as one type of stochastic programming problem, we made progress on a variant of the secretary problem, developing a method that gives an expected constant rank of the hired secretary independent of the number of candidates.

In the classical secretary problem, n items or options are presented one by one in random order (*i.e.*, all $n!$ possible orders being equally likely). If we could observe them all, we could rank them totally with no ties, from best (rank 1) to worst (rank

n). However, when the ith object appears, we can observe only its rank relative to the previous $i-1$ objects; the relative rank is equal to one plus the number of the predecessors of i which are preferred to i. We must accept or reject each object, irrevocably, on the basis of its rank relative to the objects already seen, and we are required to select k objects. The problem has two main variants. In the first, the goal is to maximize the probability of obtaining the best k objects. In the second, the goal is to minimize the expectation of the sum of the ranks of the selected objects or, more generally, for a given positive integer z, minimize the expectation of the sum of the zth powers of the ranks.

Solutions to the classical problem apply also in variety of more general situations. Examples include (i) the case where objects are drawn from some probability distribution; the interesting feature of this variant is that the decisions of the algorithms may be based not only on the relative rank of the item but also on an absolute "grade" that the item receives, (ii) the number of objects is not known in advance, (iii) objects arrive at random times, (iv) some limited backtracking is allowed: objects that were rejected may be recalled, (v) the acceptance algorithm has limited memory, and also combinations of these situations. In addition to providing intuition and upper and lower bounds for the above important generalizations of the problem, solutions to the classical problem also provide in many cases very good approximations, or even exact solutions. Our methods can also be directly extended to apply for these generalizations.

The obvious application to choosing a best applicant for a job gives the problem its common name, although the problem (and our results) has a number of other applications in computer science. For any problem with a very large data set, it may be impractical to backtrack and select previous choices. For example, in the context of data mining, selecting records with best fit to requirements, or retrieving images from digital libraries. In such applications limited backtracking may be possible, and in fact this is one of the generalizations mentioned above. Another important application is when one needs to choose an appropriate sample from a population for the purpose of some study. In other applications the items may be jobs for scheduling, opportunities for investment, objects for fellowships, etc.

The problem has been extensively studied in the probability and statistics literature.

Consider the simple case where only one object has to be selected. Since the observer cannot go back and choose a previously presented object which, in retrospect, turns out to be the best, it clearly has to balance the risk of stopping too soon and accepting an apparently desirable object when an even better one might still arrive, against the risk of waiting for too long and then find that the best item had been rejected earlier.

It is easy to see that the optimal probability of selecting the best item does *not* tend to zero as n tends to infinity; consider the following stopping rule: reject the first half of the objects and then select the first relatively best one (if any). This rule chooses the best object whenever the latter is among the second half of the objects while the second best object is among the first half. Hence, for every n, this rule succeeds with probability greater than $1/4$. Indeed, it has been established that there exists an optimal rule that has the following form: reject the first $r-1$ objects and then select the first relatively best one or, if none has been chosen through the end, accept the last object. When n tends to infinity, the optimal value of r tends to n/e, and the probability of selecting the best is approximately $1/e$.

It is not as easy to see that the optimal expected rank of the selected object tends to a finite limit as n tends to infinity. Observe that the above algorithm (for maximizing the probability of selecting the best object) yields an expected rank of $n/(2e)$ for the selected item; the argument is as follows. With probability $1/e$, the best item is among the first n/e items, and in this case the algorithm selects the last item. The conditional expectation of the rank of the last object in this case is approximately $n/2$. Thus, the expected rank for the selected object in this algorithm tends to infinity with n. Indeed, in this paper we show that, surprisingly, the two goals are in fact in conflict.

It can be proven by backward induction that there exists an optimal policy for minimizing the expected rank of selected item that has the following form: accept an object if and only if its rank relative to the previously seen objects exceeds a certain threshold (depending on the number of objects seen so far). Note that while the optimal algorithm for maximizing the probability of selecting the best has to remember only the best object seen so far, the threshold algorithm has to remember all the previous objects.

There has been much interest in the case where more than one object has to be selected. It is not hard to see that for every fixed k, the maximum probability of selecting the best k objects does not tend to zero as n tends to infinity. For the case where k is general, it was shown that there is an optimal policy with the following threshold form: accept an object with a given relative rank if and only the number of observations exceeds a critical number that depends on the number of items selected so far; in addition, an object which is worse than any of the already rejected objects need not be considered. Notice that this means that not all previously seen items have to be remembered, but only those that were already selected and the best among all those that were already rejected.

In analogy to the case of $k = 1$, bounding the optimal expected sum of ranks of k selected items appears to be considerably harder than minimizing the probability of selecting the best k items. Also, here it is not obvious to see whether or not this sum tends to a finite limit when n tends to infinity. Backward induction gives recurrences that seem even harder to solve than those derived for the case of maximizing the probability of selecting the best k.

Thus, the question of whether the expected sum of ranks of selected items tends to infinity with n has been open. There has not been any explicit solution for obtaining a bounded expected sum. Thus the second, possibly more realistic, variant of the secretary problem has remained open.

In [3] we presented a family of explicit algorithms for the secretary problem such that for each positive integer z, the family includes an algorithm for accepting items, where for all values of n and k, the resulting expected sum of the zth powers of the ranks of the accepted items is at most $k^{z+1}/(z+1) + C(z) \cdot k^{z+0.5} \log k$, where $C(z)$ is a constant. Clearly, the sum of ranks of the zth powers of the best k objects is $k^{z+1}/(z+1) + O(k^z)$. Thus, the sum achieved by our algorithms is not only bounded by a value independent of n, but also differs from the best possible sum only by a relatively small amount. For every fixed k, this expected sum is bounded by a constant. Thus we resolve the above open questions regarding the expected sum of ranks and, in general, zth powers of ranks, of the selected objects.

Our approach is very different from the dynamic programming approach taken in most of the papers mentioned above. In addition to being more successful in obtaining explicit solution to this classical problem, it can more easily be used to obtain explicit solutions for numerous generalizations, because it does not require a completely new derivation for each objective function.

We remark that our approach does not partition the items into k groups and select one item in each. Such a naive method is suboptimal. Since the expected sums achieved by our algorithms depend only on k and z and, in addition, the probability of our algorithms to select an object does not decrease with its rank, it will follow that the probabilities of our algorithms to actually select the best k objects depend only on k and z, and hence for fixed k and z, do not tend to zero when n tends to infinity.

In contrast, for any algorithm for the problem, if the order of arrival of items is the worst possible (*i.e.*, generated by an oblivious adversary), then the algorithm yields an expected sum of at least $kn^z 2^{-(z+1)}$ for the zth powers of the ranks of selected items. Our lower bound holds also for randomized algorithms.

6.

M. Ajtai obtained a very interesting complexity result as follows. The main result is a proof of the theorem that the problem of finding in a lattice a vector of minimum L_2-norm is NP-hard in randomized reductions. (A problem P is NP-hard for random reductions if every problem in NP can be reduced to a polynomial number of instances of the problem P by a polynomial time probabilistic algorithm.) It was known that the shortest vector problem in L_1 and L_∞ is NP-hard (for deterministic reductions), and the NP-hardness of the most natural case the case of L_2 was conjectured (by Van Emde Boas) already almost twenty years ago. Although NP-hardness for randomized reduction is a somewhat weaker concept then for deterministic reductions, still it provides convincing evidence that the problem is computationally infeasible. It implies that if the problem has a probabilistic polynomial time solution then every problem in NP-has such a solution. (There are relatively few interesting problems whose NP-hardness was proved for randomized reductions but not for deterministic ones. Adleman has shown that factoring integers can be reduced this way to the shortest vector problem by assuming certain unproven number-theoretical hypotheses. This work has started as an attempt to get the same result without any unproven assumptions. Finding a short vector in a lattice is one of the most basic questions in optimization problems and has many applications including integer programming, cryptography, factoring algorithms for polynomials, etc.

The shortest vector problem is the following: given a lattice L in the n-dimensional Euclidean space find a shortest non-zero vector in it. We prove that even an approximate version of this problem is NP-hard for randomized reduction, namely we show that there is an absolute constant $c > 0$ so that the problem "find a vector $v \in L$, $v \neq 0$ so that if v_0 is a shortest non-zero vector then $\|v\| \leq (1 + 2^{-n^c})\|v_0\|$" is NP-hard for randomized reduction. Our theorem also implies that the problem find a vector in L whose length is less then w, where w is a integer is NP-complete for randomized reductions. We prove the result by showing first that in a certain lattice defined by from the logarithms of small primes (an extension of the lattice that Adleman used in his proof) each short vector has only $0, 1$ coefficients in a suitably chosen basis. This creates a connection between the shortest vector problem and the subset sum problem. Through several steps we make this connection closer and closer and finally we reduce the subset sum problem, which is known to be NP-complete, to the shortest vector problem. A paper titled "The shortest vector problem in L_2" is now in preparation.

Ajtai worked with C. Dwork on a public key crypto-system with average-case worst-case equivalence. The goal of this work is to show that certain very natural lattice-problems that occur frequently in integral optimization and are important for cryptographic applications, are as difficult in the worst-case as in the average case. The result makes it possible to create individual instances of problems that are just as difficult as the worst-case problem, which makes the cryptographic protocols based on them more secure. It also shows that certain basic optimization problems are computationally infeasible even in the average case (provided that their worst-case version is infeasible). In most cases the average case problems are more realistic, they are closer to the problems occurring in applications then the worst-case problems, however the widely accepted conjectures/assumptions always concern the difficulty of worst-case problems. Our result therefore widens the scope of these assumptions to a class of problems with greater significance.

The unique shortest vector problem is the following: Assume that an n-dimensional lattice L is given so that it has a unique shortest non-zero vector v in the sense that any other vector u which is at most n^c times longer than u is parallel to v, where c is a fixed constant. Find v if L is represented by an arbitrary basis. This problem as a worst-case problem is considered computationally difficult (for an arbitrary but fixed c) although it is not known whether it is NP-complete. We show that if this problem has no polynomial probabilistic time solution (for some $c > 5$) then this is also true for the following average-case problem. Assume that an infinite set of equidistant hyperplanes is given in the n-dimensional space. (That is all of the hyperplanes defined by equations $u \cdot x = a$ where $u \neq 0$ is a fixed vector and a takes every possible integer value). Let H be the union of the hyperplanes inside a large (compared to the distance d of the neighboring hyperplanes) cube Q. We take a polynomial number of random points on H then perturb them slightly (to a distance less than $n^{-5}d$). We get the points $p_1, \ldots, p_m$ that are not on the hyperplanes any more just close to them. We show that if the worst-case unique shortest vector problem has no polynomial time solution then it is not possible to reconstruct the hyperplanes (even in an approximate sense) knowing only the points $p_1, \ldots, p_m$. Actually we prove more, we show that the sequence $p_1, \ldots, p_m$ is computationally indistinguishable from a sequence of random points chosen with uniform distribution from Q. This leads to the construction of a random number generator and other cryptographic applications. It also shows that the following the optimization problem minimize $x \cdot p_i - x_i$ subject to the constraints, $x \in R^n$, $\frac{1}{2}\|x\| \leq 1$, $x_1, \ldots, x_m$ are integers is computationally infeasible even if $p_1, \ldots, p_n$ are generated at random with the described distribution. This problem is very natural, it is connected to the task of reconstructing a set of linear functions from their approximate values. It also has an inherent symmetry which helps making the connection between the worst-case and average-case problems. (One infinite set of equidistant hyperplanes can be transformed by a rotation and a multiplication by a constant to any other.)

7.

During 1996 M.A. Saunders and J.A. Tomlin carried out research on numerical stability and behavior of interior point methods for regularized linear and quadratic programs.

The first phase of this research was the investigation of stable methods for reducing the Newton systems characteristic of modern interior methods to Karush-Kuhn-Tucker (KKT) systems. Several alternative methods were proposed and numerically tested. These results are described in [9]

The second phase of this research investigated the efficiency and stability of reduced KKT methods for regularized LP's when some essential parameters are varied. The concern is to have the regularization parameters large enough to ensure stability, but not so large as to excessively perturb the original problem. The extent of reduction of the KKT system also can have a significant effect on both stability and efficiency. These results are described in [10].

8.

In [7] we used a novel application of linear programming to image processing, where we successfully separated background embedded in objects approximating the background by low order polynomial using a metric that combined L_1 and L_∞.

References

[1] M. Kojima, N. Megiddo, S. Mizuno and S. Shindoh, "Decomposition in Interior-Point Methods," Research Report RJ 9901, IBM Almaden Research Center, San Jose, California, 1994.

[2] D. Koller, N. Megiddo and B. von Stengel, "Efficient computation of equilibria for extensive two-person games," *Games and Economic Behavior*, **14** (1996) 220–246; also "Fast algorithms for finding randomized strategies in game trees," in: Proceedings of 26th Annual ACM Symposium on Theory of Computing (1994), ACM, New York, 1994, pp. 750–759.

[3] M. Ajtai, N. Megiddo and O. Waarts, "Improved algorithms and analysis for secretary problems and generalizations," in: Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (1995), IEEE Computer Society Press, Los Angeles, 1995, pp. 473–482.

[4] N. Megiddo, S. Mizuno and T. Tsuchiya, "A linear programming instance with many crossover events," *Journal of Complexity* **12** (1996) 474–479.

[5] N. Megiddo, "Finding a line of sight through boxes in d-space in linear time," Research Report RJ 10018, IBM Almaden Research Center, San Jose, California, 1996.

[6] S. Mizuno, T. Tsuchiya and N. Megiddo, "A modified layered-step interior-point algorithm for linear programming," Research Report 10028, IBM Almaden Research Center, San Jose, California, 1996.

[7] Q. Huang and N. Megiddo, "Color image background segmentation and representation," in: Proceedings of the 1996 IEEE International Conference on Image Processing (ICIP'96, September 16–19, 1996, Lausanne, Switzerland), IEEE Signal Processing Society, 1996.

[8] M. Ajtai and C. Dwork, "A public key cryptosystem with worst-case average-case equivalence," Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, May 1997.

[9] M.A. Saunders and J.A. Tomlin, "Stable reduction to KKT Systems in Barrier Methods for Linear and Quadratic Programming," Research Report RJ 10039, IBM Almaden Research Center, San Jose, California, 1996.

[10] M.A. Saunders and J.A. Tomlin, "Solving Regularized Linear Programs Using Barrier Methods and KKT Systems," Research Report RJ 10064, IBM Almaden Research Center, San Jose, California, 1996.